AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
ADDING VALUE TO ENGINEERING

A PRELIMENERY REPORT ON

"AUTOMATIC TIMETABLE GENERATOR USING GENETIC ALGORITHM "

SUBMITTED TO THE SAVITRIBAI PHULE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Mr. PRANAV SAYKAR	Exam No. 72024105H
Mr. PRATIK LANDGHULE	Exam No. 72023950J
Mr. PARIKSHIT SHINDE	Exam No. 71908217L
Mr. TANMAY JANRAO	Exam No. 72023889H

Under The Guidance of

Mrs.A.G.Said

DEPARTMENT OF COMPUTER ENGINEERING

ALL INDIA SHRI SHIVAJI MEOMRIAL SOCIETY'S
INSTITUTE OF INFORMATION TECHNOLOGY
KENNEDY ROAD, NEAR R.T.O. PUNE-411001

SAVITRIBAI PHULE PUNE UNIVERSITY

2022-2023

AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
(IOIT)
ADDING VALUE TO ENGINEERING
An Autonomous Institute Affiliated to Savitribai Phule Pune University
Approved by AICTE, New Delhi and Recognised by Govt. of Maharashtra
Accredited by NAAC with "A+" Grade | NBA - 5 UG Programmes

CERTIFICATE

This is to certify that the project report entitles

"AUTOMATIC TIMETABLE GENERATOR USING GENETIC ALGORITHM "

Submitted by

Mr. PRANAV SAYKAR Exam No. 72024105H
Mr. PRATIK LANDGHULE Exam No. 72023950J
Mr. PARIKSHIT SHINDE Exam No. 71908217L
Mr. TANMAY JANRAO Exam No. 72023889H

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Mrs.A.G.Said** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degrees of **Bachelor of Engineering** (Computer Engineering).

(Mrs.A.G.Said)
Guide
Department of Computer Engineering

(Dr. S.N. Zaware)
Head
Department of Computer Engineering

(Dr. P.B. Mane)
Principal
AISSMS Institute of Information Technology, Pune-411001

Place: PUNE
Date:

ACKNOWLEDGMENT

It gives us immense pleasure in presenting the project report on **"AUTOMATIC TIMETABLE GENERATOR USING GENETIC ALGORITHM"** The success and the outcome of this report required a lot of guidance. We are very grateful to our guide **Mrs.A.G.Said** who has provided expertise and encouragement. We thank sir who provided vision and knowledge that was very helpful throughout the research. All that we have done is only due to the great guidance. We also express our gratitude to **Dr. S.N. Zaware** Head of Computer Engineering Department, AISSMS's Institute of Information Technology, for the valuable support.

We would also like to extend our sincere thanks to Principal Dr.P.B. Mane, for his dynamic and valuable guidance throughout the project and providing the necessary facilities that helped us to complete our dissertation work.

Mr. Pranav Saykar
Mr. Pratik Landghule
Mr. Parikshit Shinde
Mr. Tanmay Janrao

ABSTRACT

Timetable generation is a difficult task but it is very important in educational institutions. Currently timetable is managed manually. It will help to manage all the periods automatically and also will be helpful for faculty to get timetable in their phone by using application. It will also manage timetable when any teacher is absent, late coming or early going. Maximum workload for a Faculty for a day, week and month will be specified for the efficient generation of timetable .To create timetable it requires lots of time and man power .In our paper we have tried to reduce these difficulties of generating timetable by genetics algorithm. The system administrator logs into the system and then the administrator input the courses with their codes and the unit. After inputting the courses, it moves to the next page where all the lecture halls or rooms that will be used will be inputted. After inputting these, the system then generates the timetable system. This technique (genetic algorithm) used helps in reducing to minimum errors and mistakes in encountered in developing an automatic timetable. By using genetic algorithm we are able to reduce the time require to generate time table and generate a timetable which is more accurate, precise and free of human errors. The first phase contains all the common compulsory classes of the institute, which are scheduled by a central team. The second phase contains the individual departmental classes. Presently this timetable is prepared manually, by manipulating those of earlier years, with the only aim of producing a feasible timetable.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Overview

In this proposed system, we are implementing a Automatic timetable generator using genetic algorithm which create a timetable automatically and accurately in small amount of time .

1.2 Motivation

Automatic timetable generator using genetic algorithm is A time table ensures that each class has only one teacher during learning hours of a particular period. Time table also reduces the confusion while learning. students are very clear about the subject they have to study in a particular period.

- Providing a automatic time table generator will help to generate time table automatically
- Proposed system of our project will help to generate it automatically also helps to save time.
- It avoids the complexity of setting and managing timetable manually
- Saves on effort and time.
- Provides a sense of security in that you had planned on what to learn.
- It makes you accountable.
- The motivation here is to build such tool which is computationally efficient and creates summaries automatically.

1.3 Problem Definition and Objectives

Normally timetable generation is done manually. as we know, all institutions have their timetable, managing and maintaining these will not be difficult. Considering workload with this scheduling will make it more complex. Generally, while generating a timetable, consideration should be given to the maximum and least workload that a college is exposed to.Timetable generation gets more complicated in certain instances. Also, it is a time-consuming process.

1.4 Project Scope and Limitations

This software is a solution for the time table generation problem manually. It's main scope is to save the time and efforts for time table generation process. The data of faculty in the data base can further be used to maintain record of faculty's experience for particular subjects . The future enhancement that can be developed from the project is to generate the master timetable for the departments and to the entire college. This enhancement can be achieved my making further modifications keeping the approach and techniques used in this project.

1.5 Methodologies of Problem Solving

Our system is implemented in different modules:

1. Requirement Analysis:
   Begin by analyzing the requirements and constraints of the automatic timetable generator project. This involves understanding the desired features, user expectations, input data sources, and any specific limitations or rules that need to be followed. Clear requirement analysis helps in defining the scope of the project and guides the development process.

2. User-Centered Design:
   Adopt a user-centered design approach to ensure the timetable generator meets the needs of its intended users. This involves conducting user research, creating user personas, and designing user interfaces that are intuitive, easy to navigate, and visually appealing. Usability testing and feedback collection are also important to iteratively improve the user experience.

3. Backend Development:
   Utilize backend development methodologies to implement the logic and algorithms for generating the timetable. This could involve designing and implementing data models, creating APIs for data retrieval and manipulation, and developing algorithms or rule engines for scheduling courses, instructors, and rooms. Backend frameworks like Sql, Php on Rails can be utilized for efficient development.

4. Frontend Development:
   Apply frontend development methodologies to create the user interface and

interactivity of the timetable generator. Use HTML, CSS, and php to design and implement responsive and user-friendly interfaces. Adopt modern frontend frameworks like React to streamline the development process and enhance user experience.

5. Testing and Quality Assurance:
Incorporate testing methodologies throughout the development process to ensure the functionality, reliability, and performance of the timetable generator. Implement unit testing, integration testing, and end-to-end testing to identify and fix any issues or bugs. Additionally, conduct user acceptance testing to validate that the system meets the specified requirements.

6. Version Control:
Employ version control methodologies, such as Git, to manage the source code and facilitate collaboration among team members. Use branching and merging strategies to maintain a clean codebase, track changes, and manage different development stages effectively. requirements.

7. Documentation:
Document the project using appropriate methodologies to ensure clarity and ease of maintenance. This includes writing technical documentation, API documentation, user manuals, and providing code comments to enhance code readability and understanding.

CHAPTER 2

LITERATURE SURVEY

J.J Grefenstette [1] evolutionary approaches to the time scheduling problem. styles comparable to the evolutionary algorithm set and genetic algorithms. mixed results while in usage. In this essay, we've examined the challenge of using a genetic algorithm to catalogue an instructional timeline. A synthetic inheritable defensive network and a non-original mongrel algorithm were also used to solve the issue, and the outcomes were compared to those obtained using the inheritable algorithm. The findings demonstrate that GAIN is more suitable than GA for arriving at a potential conclusion quickly

Srinevasan, Dipti [2], Academics continually struggle with the issue of discovering the study schedule that is feasible at the university's major department. This research provides an evolutionary algorithm (EA) technique built on solving the resilient schedule problem at the university. On to chromosomal representation issues. At the correct computer moment, heuristics and contextually grounded thinking may have been attained. Cohesion has been improved by the clever usage of inheritable revision. Using actual data from a top university, the full class plan described in this paper is accepted, estimated, and discussed. An automatic timeline system that.

Anuja Chowdhary [3] Devloped uses an efficient timing algorithm that can handle both strong and weak barriers. After a specific semester is over, each teacher and student can look at their timetable, but they shouldn't plan ahead. According to the teacher's schedule, the availability and power of visual resources, and other rules relevant to various courses, semesters, preceptors, and grade positions, the Timetable generator system develops a timeline for each class and teacher.

Anirudha Nanda [4] Proposes a typical resolution to the timing issue. most difficult past heuristic programmes that have been suggested from a scholarly perspective. This outcome is still valid when viewed from the perspective of the topic, specifically the teacher's inadequacy at a certain moment. The planning result described in this work is adaptable, with the main goal of resolving academic and academic dispute, schoolteacher-related concerns. All implicit walls (e.g., schoolteacher vacuity, etc.) are replied firmly.

Al-Khair [5] Suggested using algorithmic techniques to solve the timing issue with teacher availability admissions. The challenge of scheduling academy time is completely addressed by this technique, which employs a heuristic approach. First, it creates a temporary timeline using randomly generated title sequences. If a teacher divides a class into more subjects than are allowed, the extra subjects are moved to the Conflict data structure. By taking into account the tight relation-

ship between the schedule problem and the vertex colouring problem,
Csima and Gotlieb[6] also addressed the schedule problem as a three-dimensional assignment problem. The now wellknown particular link between the diverse scheduling difficulties was originally examined in this study. Becker and Baraclough both deceived their respective executions with "hand" computations, extending the work of Csima and Gotlieb. These publications typically relied on a heuristic methodology. This work led to a large number of papers that examined the issue but didn't add any fresh insight..
Broder [7] asserted that the objective function can be minimised by repeatedly assessing a collection of relevant nonlinear equations (deduced from arbitrary or montecarlo assignments). It was asserted that the outcomes thus obtained might not be ideal but rather would be comparable to the initial minimal outcomes obtained by prior heuristic methods.
The original work of Kuhn's Hungarian system to the matrix reduction problem is described by Lions [8] as being crucial for the functionality of Gotlieb's set partitioning system of schedule generation. However, the first algorithm itself experienced duplicated effectiveness as a result of this. Lions added a conception system for the creation of class/teacher calendars to this basic task. In his paper, he discussed how to apply Kuhn's Hungarian system to the matrix reduction issue needed by Gotlieb's schedule generation method.
Walker and Macon [9] They implemented a Monte Carlo algorithm that randomly chooses classes for students, illustrating the difficulty of placing students in a set of courses. They observed that the application of the randomising procedure had a limit on the ultimate best outcome.
Punter [10] His method entailed transferring this data onto rendering wastes before it was processed by a computer, as well as manually producing allocation plans that outlined the instructional circumstances for each class. The discovery of contradictory sub-sets of assignment conditions was made possible by analysis employing a graph colouring approach. It was claimed that the system might be used to generate workable calendars with the appropriate emendations
Hulskamp[11], who created software to assist engineering students in investigating a wide range of potential solutions to a design problem. The software was put into use at Caulfield Institute of Technology, and it was successful in illuminating the problem's complexity and phase space. This may be the first publication of a DSS for the creation of schedule results.
Breslaw [12] created a programme that was not based on integer programming

but was also effective and effective. As a DSS, it made it simple for users and the programme to collaborate on finding the best solution. Calendars were successfully constructed (as opposed to generated) according to empirical tests.

Knauer [13] He presented a program that analysed the results and used the exchange routine for an improvement within certain limits. The algorithm was then applied to school timetables and a considerable improvement in timetable quality was achieved. These computer generated timetables were used in some of the larger high schools of Wurzburg. Again, however, no reference to the intractability of the general problem was discussed.

Miles and Roger [14] discussed issues directly connected to the organisation and related activities common to computer timetable generation. Further Egner gave a general review of the evolution of the use of the computers in the construction of school timetables in Great Britain.

Tillet [15] proposed an operations research model which discussed the feasibility of determining the optimality of an algorithm in generating a solution to the problem. The results indicated that there was improvement in only fifty percent of the cases when applying the model. It was shown that there is considerable merit in taking a quantitative optimisation approach to the problem. Such fine tuning to speed up solution generation was just starting to be considered by a number of researchers.

Harwood and Lawless[16] discussed the achievement of organisational goals using mathematical models (mixed integer programming and goal programming). The goal programming related to the provision of explicit slackvariables (variables that are not mandatory but could assist in finding an optimal solution) and the mixed integer programming referred to variables, which can only take on integer values in the final solution.

Hulskamp [17]in which he developed software to assist engineering students in exploring many alternative solutions to a design problem. The software was implemented at Caulfield Institute of Technology with positive results in demonstrating the complexity and phase space of the problem. This is arguably the first publication of a DSS (Decision Support System) for timetable solution generation.

Breslaw [18] developed a program which was both efficient and effective and which did not rely on integer programming. As a DSS, it allowed easy interaction between users and the program, in determining the optimal solution.Empirical tests provided successful construction (as opposed to generation) of timetables.

Punter[19] His technique involved manually produced allocation plans that listed the lesson requirements for each class, and then transferred this data onto coding sheets before processing was carried out by a computer. An analysis by using a graph colouring approach enabled the identification of inconsistent sub-sets of lesson requirements. It was asserted that with appropriate amendments the scheme could be used to generate viable timetables. However, no follow-up paper was uncovered in the literature to support his assertion

Schmidt and Strohlein [20] produced their well known annotated bibliography of the literature. It is noteworthy that Schmidt and Strohlein predicted the generation of timetables by computer to be heavily influenced by computing devices at hand. They further expanded their prediction in stating that timetable programs will move from remote handling in huge computing centres to micro computer centres owned by schools and handled by teachers on their own desktops directly. Twenty three years later we still do not see any evidence of their prediction and timetabling is still a manual activity carried out by administrators and not by the staff or students.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

Assumptions

1. The system is designed for creating timetable and the input given from facaulty to generate a timetable.

2. The user must have Web Browser.

3. The web app requires a proper network connection.

Dependencies

IDE ,Server ,Inputs, Networks, etc.

3.2 Functional Requirements

3.2.1 System Feature 1

- To implement a User interface on the system.
- User-friendly front-end design using Cascading Style Sheets
- Strong authentication while performing various operations.
- php validations and alerts wherever needed.
- Based on the Input Data Identify it and shows output according to input and need of customer.

3.2.2 External Interface Requirements

- User Interfaces- The webapp will have take input from user and create output and show it.
- Hardware Interfaces webApp will run on platforms i.e.web.

3.3 Non Functional Requirements

3.3.1 Usability

- The system should be easy to use. The user should be able to see the result after.
- passing the image and then just by clicking the predict result Button ,output should be displayed on the screen.

3.3.2 Reliability

- Implement thorough input validation mechanisms to ensure that constraints and preferences entered by administrators are valid and consistent.
- Also, user provided data will be used to compare with result and generate it.

3.3.3 Performance

- Optimize the genetic algorithm implementation to minimize the computational complexity and improve efficiency.
- Optimize data processing and retrieval operations by using efficient database queries, indexes, and caching mechanisms.
- Design the system architecture to be scalable and capable of handling increasing loads and larger datasets.
- The system should implement the algorithm properly and accordingly as; first preprocessing algorithms then transfer learning algorithms.

3.3.4 Supportability

- The system is based Php and Web. So, problem on server side requires prior knowledge of the filed.
- Client side problems requires web technology and networking knowledge.

3.4 System Requirements

3.4.1 Software Requirements

- Programming Languages: Php , Html , Css ,Sql
- IDE: VScode

3.4.2 Hardware Requirements

- Processor: Intel i3 and above
- Operating System: Windows 10
- Ram: 4GB and above

3.4.3 Web Development Kit

- IDE: VScode , Server

3.5 Analysis Models: SDLC Model to be applied

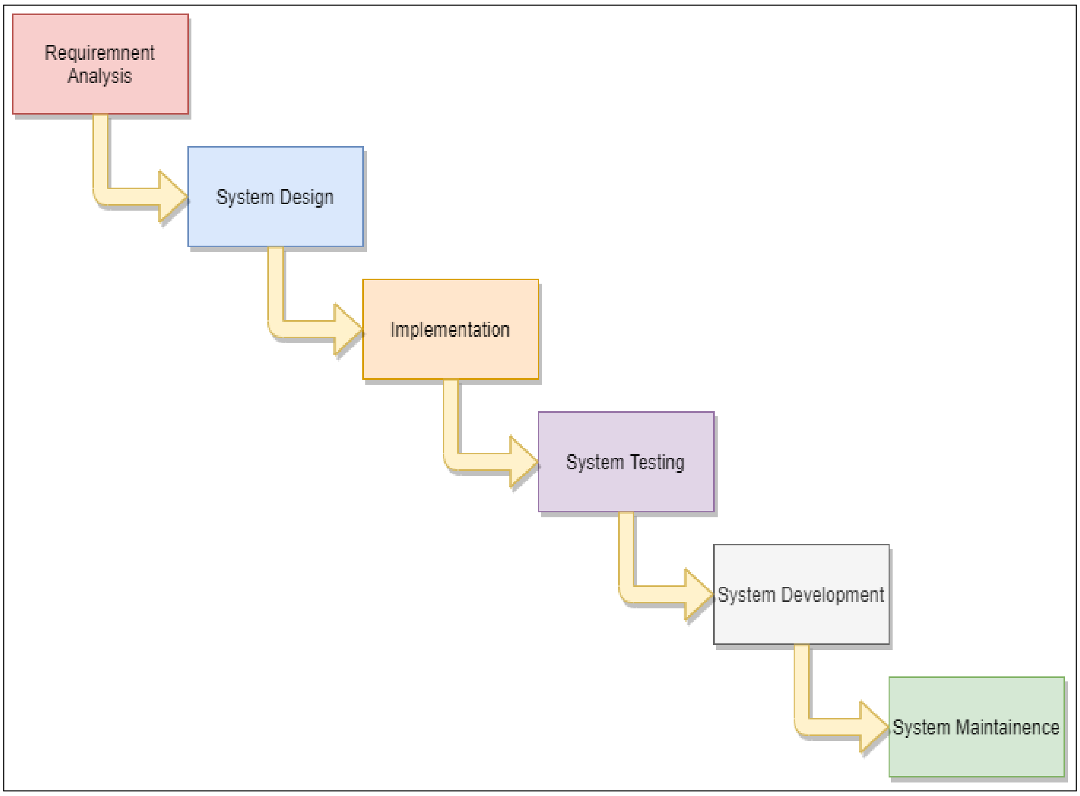


Figure 3.1: Waterfall Model

SDLC Waterfall model is being depicted by our system.

- The initial stage is requirement analysis stage here the data is being gathered which is to be provided as an input to the system.
- Second stage is the design stage where all the data is being formatted into a particular matrix. The scores are used to generate the matrix.
- Third stage is the coding stage in which the system performs its main functionality of mapping of the classes and getting the exact prototype.
- Testing is done in the fourth phase inorder to test the word with the probabilistic label.
- In the maintenance phase it's the last phase wherein the system has to depict and maintain the probabilistic labels of the respective words.

3.6 System Implementation Plan

In the system plan implementation the input can be in text, URL or file format. This data entered is first cleaned and then processed using TFIDF and text rank algorithm. After this step based on the scores generated sentences are ranked for output summary generation.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

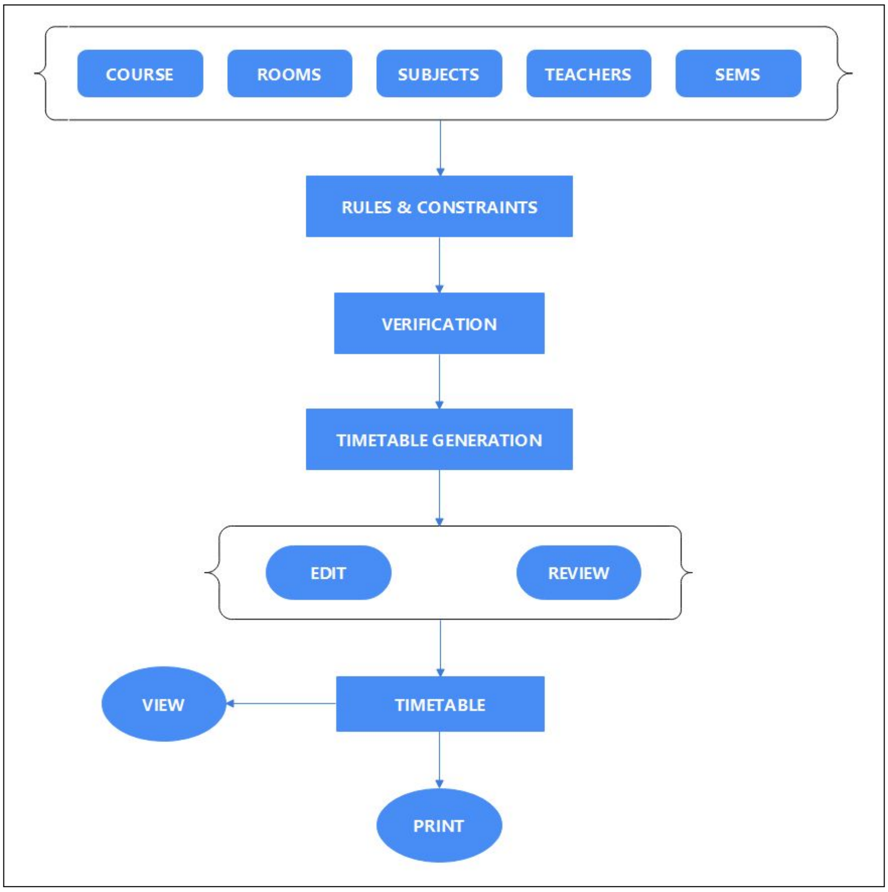


Figure 4.1: System Architecture

The client-side involves web browsers used by administrators to access the web application interface. The web browser renders HTML, CSS, and Php enabling administrators to interact with the system and input constraints, view generated timetables, and manage the scheduling process. The application layer implements the core logic and functionality of the timetable generation system. It is developed using PHP, a server-side scripting language, and includes the genetic algorithm engine, data access layer, validation and error handling, and reporting/exporting functionalities.

4.2 Data Flow (DFD) Diagrams

4.2.1 DFD Level 0 Diagram

This diagram shows the high-level interaction between the Administrator, the Automatic Timetable Generator, and the Timetables. The Administrator enters

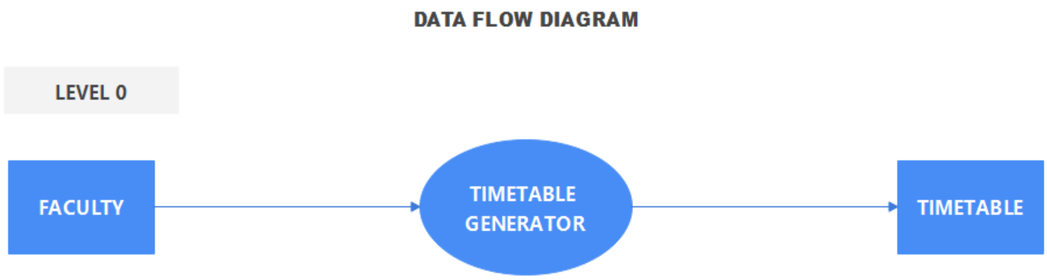


Figure 4.2: DFD Level 0 diagram of Timetable Generator

constraints and preferences, which are processed by the Automatic timetable generator to generate timetables.

4.2.2 DFD Level 1 Diagram

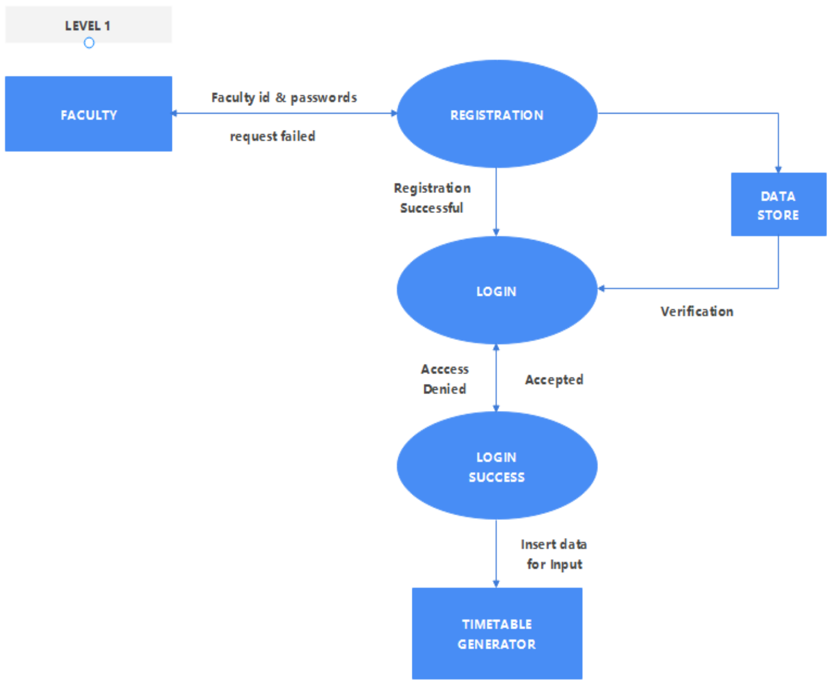


Figure 4.3: DFD Level 1 diagram of Timetable Generator

This diagram expands on the Level 0 DFD by introducing the Input System and the Output System. The Administrator interacts with the Input system to provide constraints and preferences. The input system validates the inputs and passes them to the automatic timetable generator, which generates the timetables. The output system presents the generated timetables to the Administrator.

4.2.3 DFD Level 2 Diagram

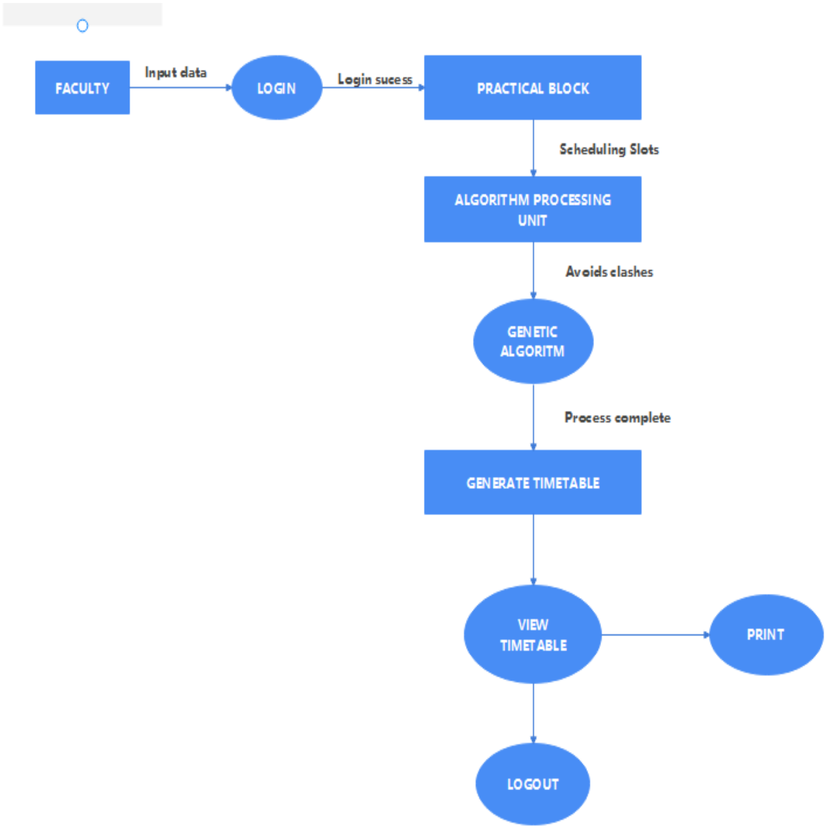


Figure 4.4: DFD Level 2 diagram of Timetable Generator

This diagram provides a more detailed view of the system by decomposing the Input System, Automatic Timetable Generator, and Output System into sub-components. The Input System includes the Data Processor, which validates and processes the inputs before passing them to the Genetic Algorithm Engine. The ggenetic algorithm engine generates the timetables based on the processed inputs. The output system is responsible for presenting the generated timetables to the administrator.

4.3 UML Diagrams

4.3.1 Activity diagram

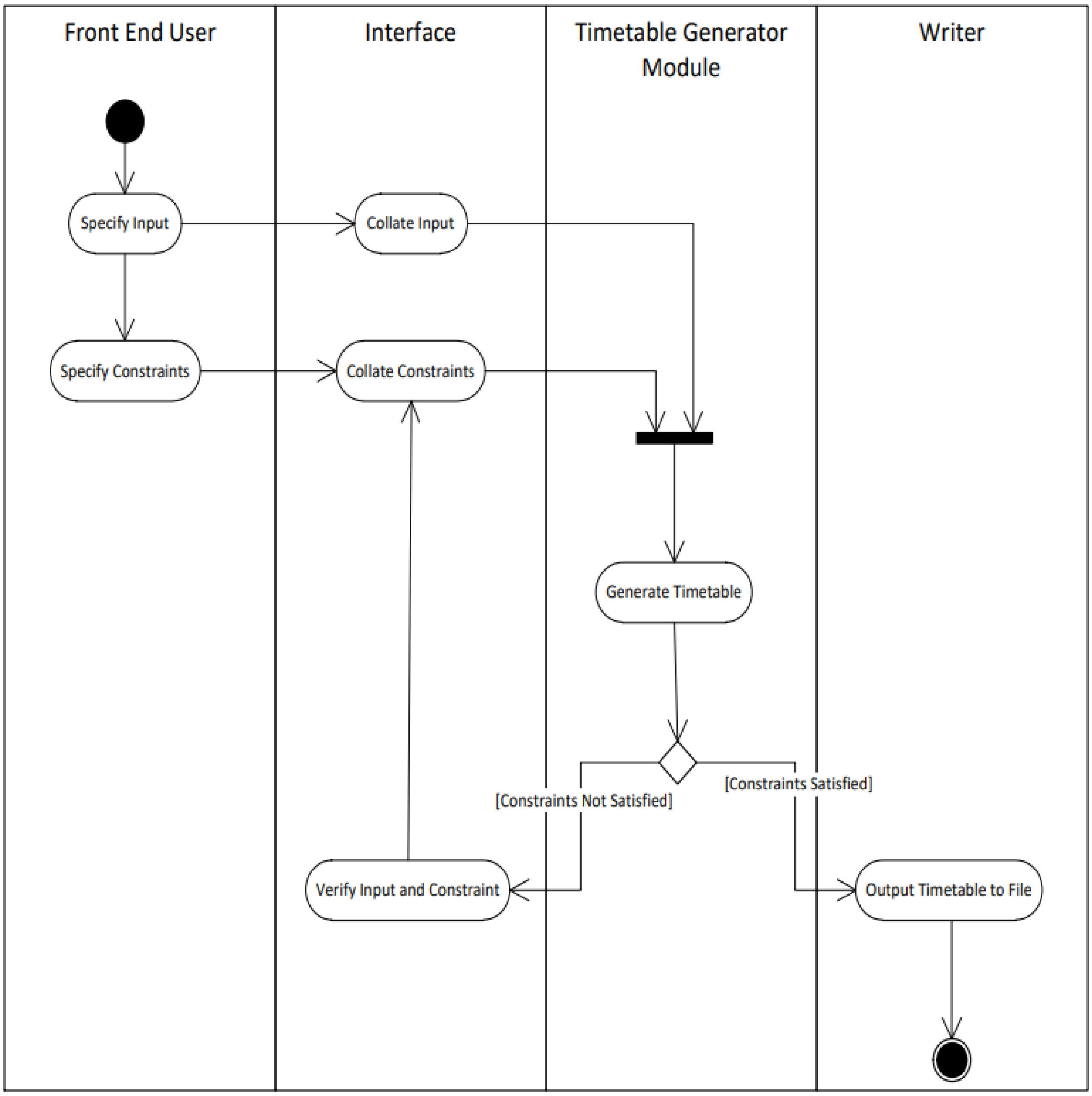


Figure 4.5: Activity diagram of Timetable Generator

- Activity Diagram is a behavioral diagram presenting the actors their functions performed.
- If the Administrator selects the Constraints option, they are directed to the Constraint Input page, where they can input various constraints such as class timings, room availability, and teacher preferences.

- The represent the individual lane as their entire activities and the functionality carried out by that particular actor in the respective lanes.
- The Administrator may choose to export the generated Timetables or generate reports summarizing the scheduling outcomes.
- Input is being provided by the user and the output is being given back to the user.

4.3.2 Sequence Diagram

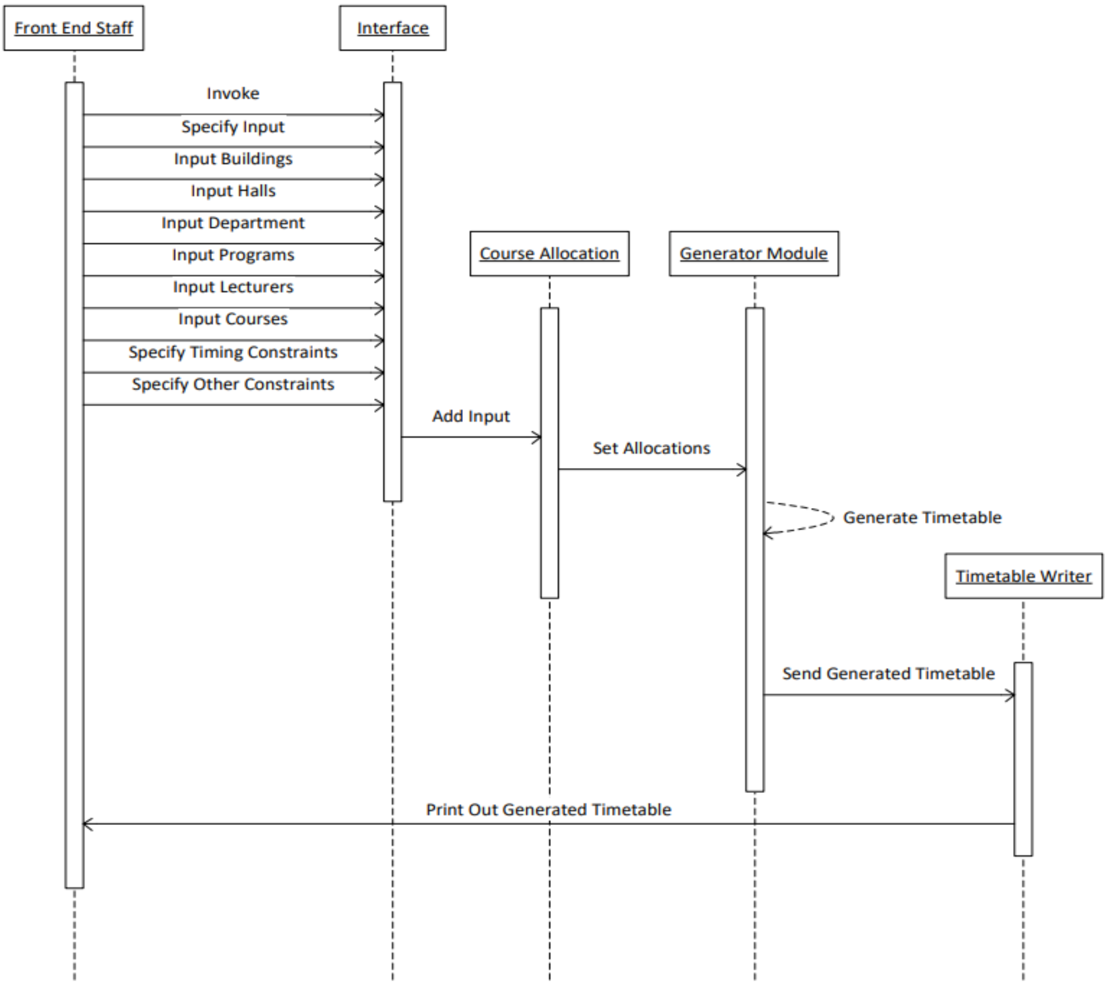


Figure 4.6: Sequence diagram of Timetable Generator

In the above Figure 4.6 there are different actors:

- Actors

- User
- Input
- Output

- Different roles played by the different actors are been specifi

  ed separately in their individual lanes.
- Each actor has number of timelines for the activity (Function) to be preformed.

4.3.3 Usecase Diagram

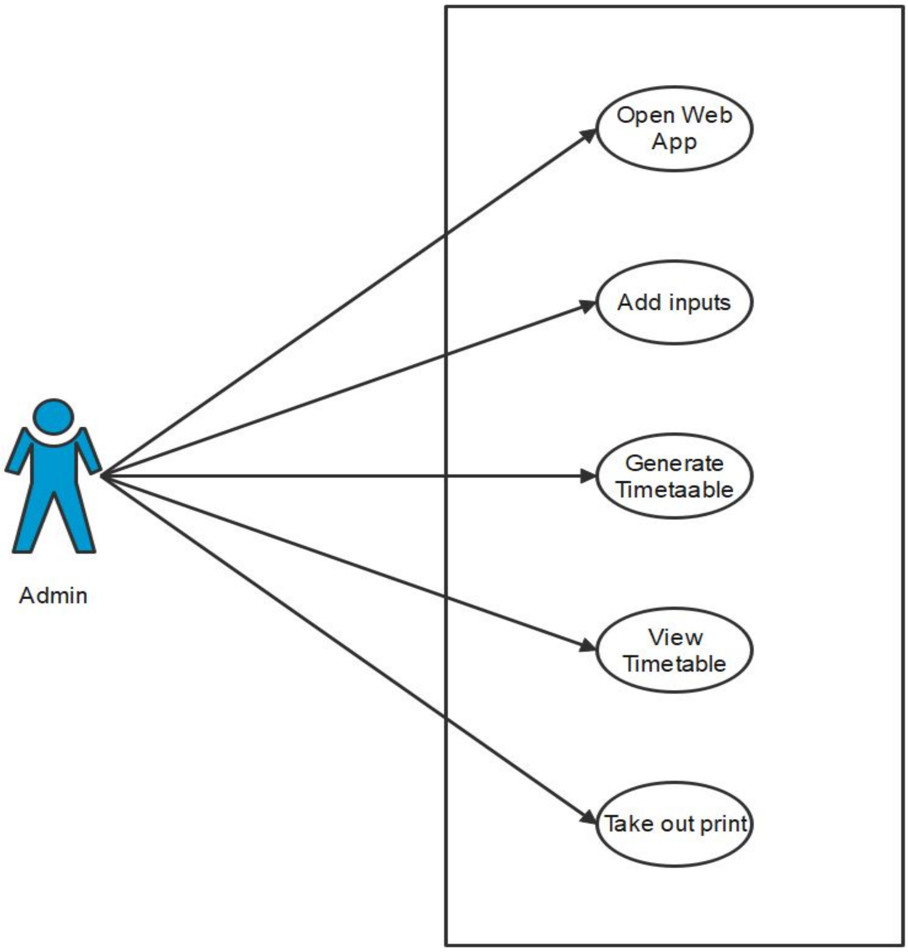


Figure 4.7: Usecase diagram of Timetable Generator

Use case diagram is used for representing the problem statement that is the actors in it, their functionality in an behavioral manner. They are useful when the system is to be in the programmatic execution.
In the above Figure 4.7 there are different actors:

- Actors
  - Admin
- Functional blocks include the stepwise execution of the entire system that is the form of different procedures (functionality).
- Input to the every function in this diagram is the output from the previous state.

CHAPTER 5

PROJECT PLAN

5.1 Project Estimate

Waterfall model is being used for the project estimation. It depicts the step wise execution of the entire project.

5.1.1 Reconciled Estimates

Cost Estimate :

Not Applicable.

Time Estimate:

Approximately eight months.

5.1.2 Project Resources

Windows System, Anaconda Navigator, VS Code, 4GB RAM, 2.93 GHZ CPU Speed.

5.2 Risk Management

The overall procedure is much optimised and time needed to generate summary is around 3 seconds. The rouge score is also better than the Creating manual timetable.

5.2.1 Risk Identification

The various risks that are identified are :

1. Need of continuous internet while running the system.
2. Need good processing power.

5.2.2 Risk Analysis

The risks are being analyzed taking into consideration the requirements for execution. The analysis include the need of internet for the project for one of the section. The need of good processing power is consideration for risk analysis because the entire computations depends on the size of input given.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Table 5.1: Risk 1 Mitigation, Monitoring, Management

Risk ID	1
Risk Description	Internet Availability
Category	Requirements
Probability	Less
Impact	High
Response	Mitigation
Strategy	Wi-Fi, LAN, Router
Risk Status	Identified

Table 5.2: Risk 2 Mitigation, Monitoring, Management

Risk ID	2
Risk Description	Need of Good Processing Power
Category	Requirements
Probability	Medium
Impact	High
Response	Mitigation
Strategy	Dataset Pre-processing
Risk Status	Identified

5.3 Project Schedule

5.3.1 Project Task Set

Major aspects of the project:

- Task 1: Correctness
- Task 2: Availability
- Task 3: Integrity

5.3.2 Timeline Chart

Table 5.3: System Implementation Plan

SR NO	DURATION	ACTIVITY PERFORMED
1	July second week 2022	Topic Finalization
2	3rd and 4th week of July 2022	Understanding of Base Paper
3	1st and 2nd week of August 2022	Literature Survey
4	3rd and 4th week of August 2022	System Architecture Design Completion
5	1stand 2nd week of September 2022	1st review Completed
6	3rd and 4th week of September 2022	2nd review Completed
7	1st week of October 2022	UML diagrams, State Charts and DFD's Completed
8	2nd week of October 2022	3rd Review Completed
9	3rd and 4th week of October 2022	Final review completed
10	November 2022	Exam
11	December second week 2022	Distribution of implementation modules
12	January 2023	Project implementation in modules
13	February 2023	Development of the entire module
14	March 2023	Testing of the project
15	April first week 2023	Final Report Submission

5.4 Team Organization

Team consists of four members. Proper planning mechanism is used and roles are analyzed and defined.

5.4.1 Team Structure

Table 5.4: Team Structure

SR No.	Member Name	Responsibility
1	Pranav Saykar	Developer , Project analysis
2	Pratik Landghule	Developer , Requirement gathering
3	Parikshit Shinde	Developer , Project Design
4	Tanmay Janrao	Developer , Testing

5.4.2 Management Reporting and Communication

Well organized plans were been made and completed accordingly within time. Progress reporting was been updated and completed. Communication as per requirements were being done effectively.

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

1. Data Collection and Preprocessing:
   In this stage, the information which irrelevant for summarization or which can affect the final result is removed.

   (a) Data Collection:
   - Course Information: Collects details about the courses offered by the educational institution, such as course code, title, credit hours, and prerequisites.
   - Course Schedule Preferences: Gathers information on preferred time slots or specific time constraints for each course.
   - Professor Information: Collects details about professors, including their names, qualifications, and expertise in specific courses.
   - Professor Availability: Captures the availability of professors based on their teaching schedules and other commitments.
   - Classroom Information: Gathers details about the available classrooms or venues for conducting classes, including the room capacity, facilities, and location.
   - Classroom Availability: Collects data on the availability of classrooms during different time slots.

   (b) Preprocessing:
   - Data Integrity Check: Ensures the correctness and consistency of the input data by validating against predefined rules and constraints.
   - Handling Missing Data: Identifies and addresses any missing or incomplete information in the input data.
   - Data Formatting: Standardizes the data format and structure to facilitate further processing.
   - Constraint Validation: Checks the validity and feasibility of constraints imposed by the educational institution, such as room capacity, professor availability, and course prerequisites.
   - Encoding and Decoding: Converts the input data into a suitable representation for the genetic algorithm, such as binary or integer encoding, to be used for creating and manipulating chromosomes.

2. Processing:

   (a) Fitness Evaluation:

      - Fitness Function Design: Defines a fitness function that quantifies the quality of a timetable solution based on predefined criteria such as minimizing clashes, optimizing room utilization, and balancing professor workloads.
      - Evaluation of Constraints: Checks the satisfaction of various constraints, including room capacity, professor availability, and course prerequisites, and assigns fitness scores accordingly.

   (b) Genetic Algorithm Operations:

      - Selection: Selects a subset of the fittest individuals (timetables) from the population for reproduction, based on their fitness scores.
      - Crossover: Performs crossover or recombination of genetic material between selected parent timetables to generate new offspring solutions.
      - Mutation: Introduces random changes or modifications to the offspring solutions to introduce diversity and avoid getting stuck in local optima.
      - Population Control: Manages the size of the population by applying strategies such as population reduction or population replenishment.

   (c) Genetic Algorithm Optimization:

      - Iterative Generation: Repeats the genetic algorithm operations (selection, crossover, mutation) for multiple generations until the termination condition is met.
      - Convergence Analysis: Monitors the convergence of the algorithm by tracking the fitness scores and other metrics across generations.

   (d) Solution Representation:

      - Chromosome Encoding: Encodes the timetable solutions into a suitable representation, such as a binary string or an array of genes, which can be manipulated by genetic operators.

- Decoding: Decodes the chromosome representation back into a human-readable format to interpret and display the generated timetables.

(e) Result Analysis and Selection:

- Timetable Selection: Identifies the best timetable(s) with the highest fitness score(s) as the final output(s) of the timetable generation process.
- Result Visualization: Presents the generated timetables in a user-friendly format, allowing users to view and analyze the schedules.

(f) Performance Optimization:

- Parallel Processing: Explores parallel computing techniques to enhance the performance and speed of the genetic algorithm operations.
- Optimization Algorithms: Implements optimization techniques (e.g., local search algorithms) in combination with the genetic algorithm to improve the efficiency and effectiveness of the timetable generation process.

3. Admin Panel:

- Manage Users: Enables the admin to view, add, edit, or delete user accounts.
- Manage Courses: Allows the admin to add, edit, or delete course information.
- Manage Professors: Provides functionality to manage professor details.
- Manage Classrooms: Allows the admin to add, edit, or delete classroom information..
- Manage Time Slots: Enables the admin to define available time slots.

4. Timetable Generation:

- Generate Timetable: Utilizes the genetic algorithm to create optimized timetables.

- Fitness Calculation: Calculates the fitness score of each timetable solution.
- Crossover: Performs the genetic operation of crossover to produce new timetable solutions.
- Mutation: Introduces random changes to the timetables for diversification.

5. Timetable Display:

   - View Timetable: Displays the generated timetables for different courses and professors.
   - Filter Options: Provides options to filter and view timetables by course, professor, or classroom.
   - Print or Export: Allows users to print or export the timetables in different formats.

6. Constraints Management:

   - Time Clash Detection: Ensures that no two courses or professors are scheduled in the same time slot.
   - Room Capacity Check: Verifies that the assigned classroom has adequate capacity for the course.
   - Professor Availability: Considers the availability of professors during scheduling.

7. Integration and Deployment:

   - Database Management: Stores and retrieves data related to users, courses, professors, and timetables.
   - Web Interface: Develops a user-friendly web interface for accessing and interacting with the application.
   - Hosting and Deployment: Deploys the application on a web server for online access.

CHAPTER 7

SOFTWARE TESTING

Software Testing is the process of executing every functionality and procedure of the program or application with the intent to find the errors or bugs. Testing is performed to investigate the entire project from every aspect. It deals with the motto to make the model more robust and accurate. The results make the developer aware of the issues that the program might go through in the future. Software testing is important to understand the future risks.

7.1 Type of Testing

7.1.1 Positive Testing

In this process the testing is done against valid data inputs. The model is being given the same kind of input which it is trained on and designed for. It only checks on the valid set of inputs. This checks whether the system actually shows error when the system is supposed to. Positive testing tries to predict whether the system shows the exact output as per the requirements and specifications. Its checks the expected behavior of the model.

7.1.2 Negative Testing

In this process the testing is done against invalid data inputs. The model is being given the different kinds of input which it is not trained on and not designed for accordingly. It only checks on the invalid set of inputs. This checks whether the system actually shows error when the system is supposed to. Negative testing tries to predict whether the system shows the exact output as per the requirements and specifications when the inputs are not in correspondence with the valid inputs. It checks the expected behavior of the model in case of random unwanted actions.

7.1.3 Unit Testing

The unit testing is performed on every part of the model that is being developed. It tests the model right from the stage wherein the input was been provided upto the state where the probabilistic labels are being generated. The tests test the inputs, outputs, functions, classes, modules and the entire data in chunks. This leads to the assurance of having no errors in the smallest possible module too. This helps in making the model more confident.

7.1.4 Integration Testing

In integration testing the entire system is being tested as a whole entity. The combination of various modules lead to the formation of an integrated system. The unit testing tests the individual elements and the integration testing tests all the unit combined as a single unit. It ensures that the functionality of the system is same and error free when combined completely.

7.2 Test cases & Test Results

Table 7.1: Test Cases and Test Results

Test Case ID	Test Case Name	Objectives	Expected Output	Actual Output	Result
1	Check College Details	Fill all the details related to the college and all fields are mandatory	To verify that details are filled or not	details are filled	Pass
2	Check subjects,teacher detail	To check that details are filled properly then form will submit	To verify that all details are correct or not	All details are correct	Pass
3	Check Generate timetable option	To check the button is working or not properly	Summary should be generated	Timetable is generated	Pass
4	Check Print option	To check that pdf is generared or not	pdf should be generated	pdf is generated	Pass

CHAPTER 8

RESULTS

8.1 Screenshots

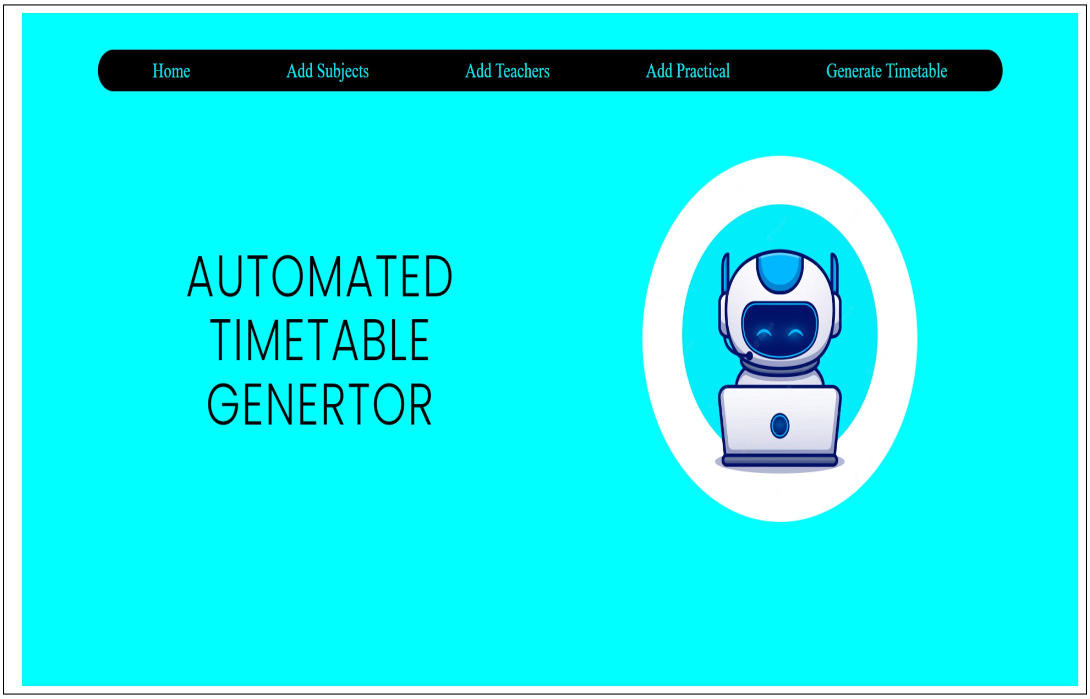


Figure 8.1: Output 1

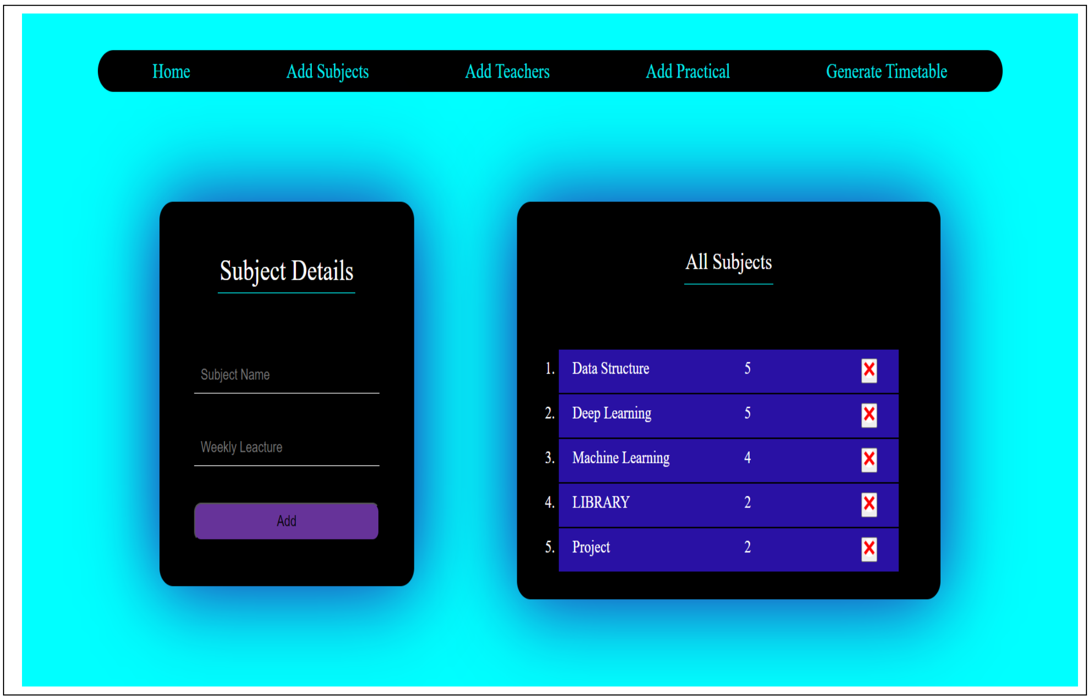


Figure 8.2: Output 2

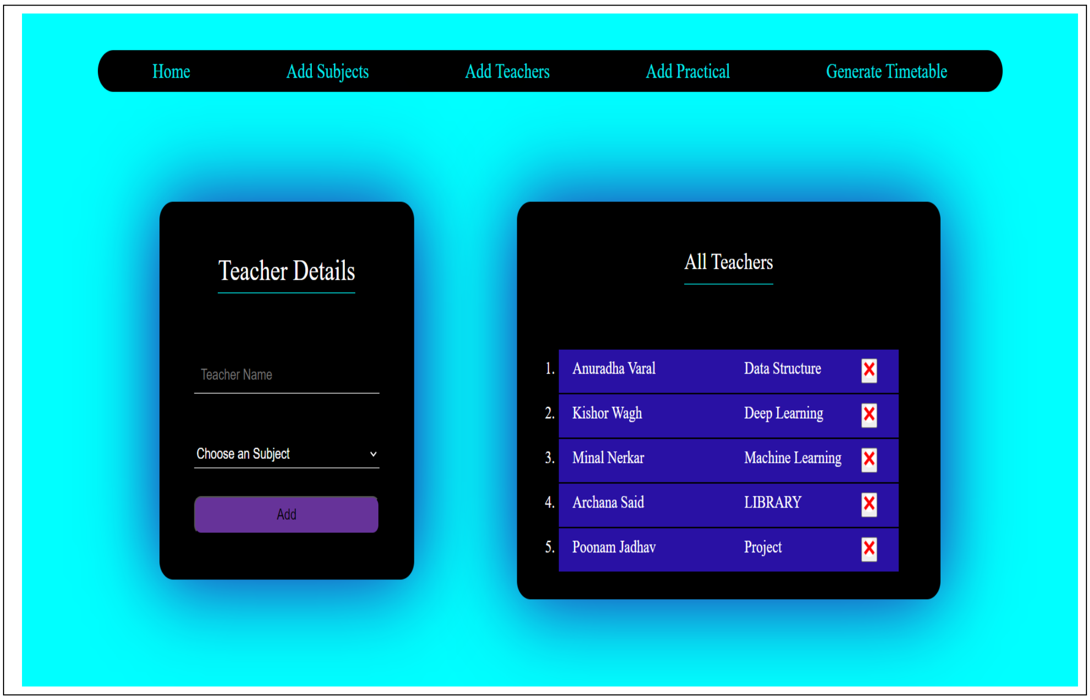


Figure 8.3: Output 3

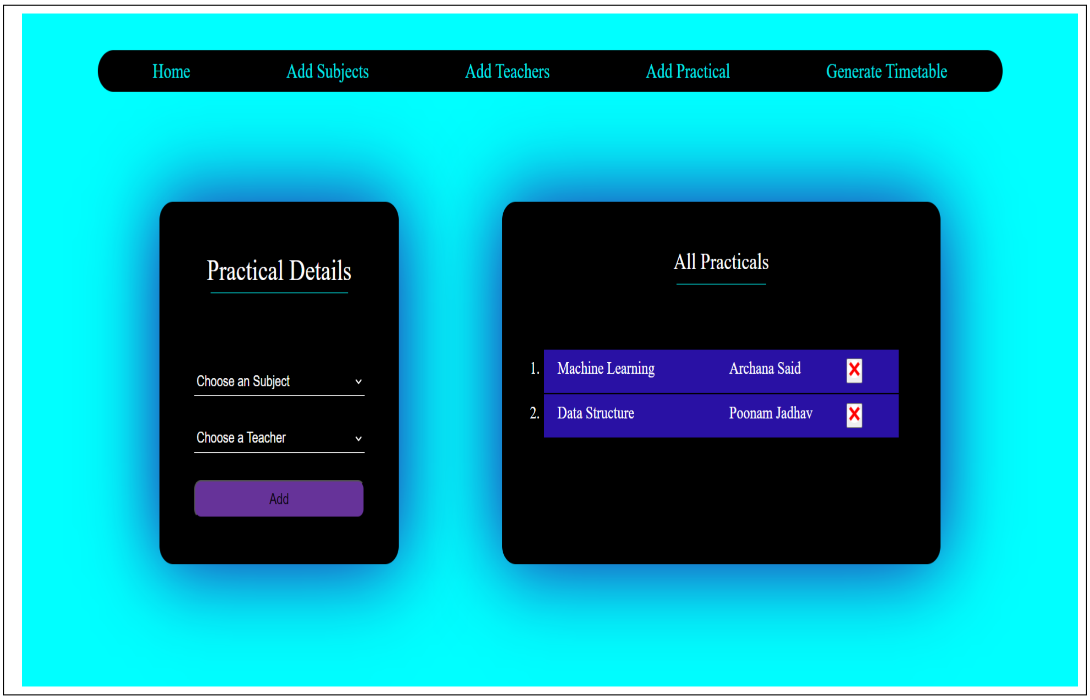


Figure 8.4: Output 4

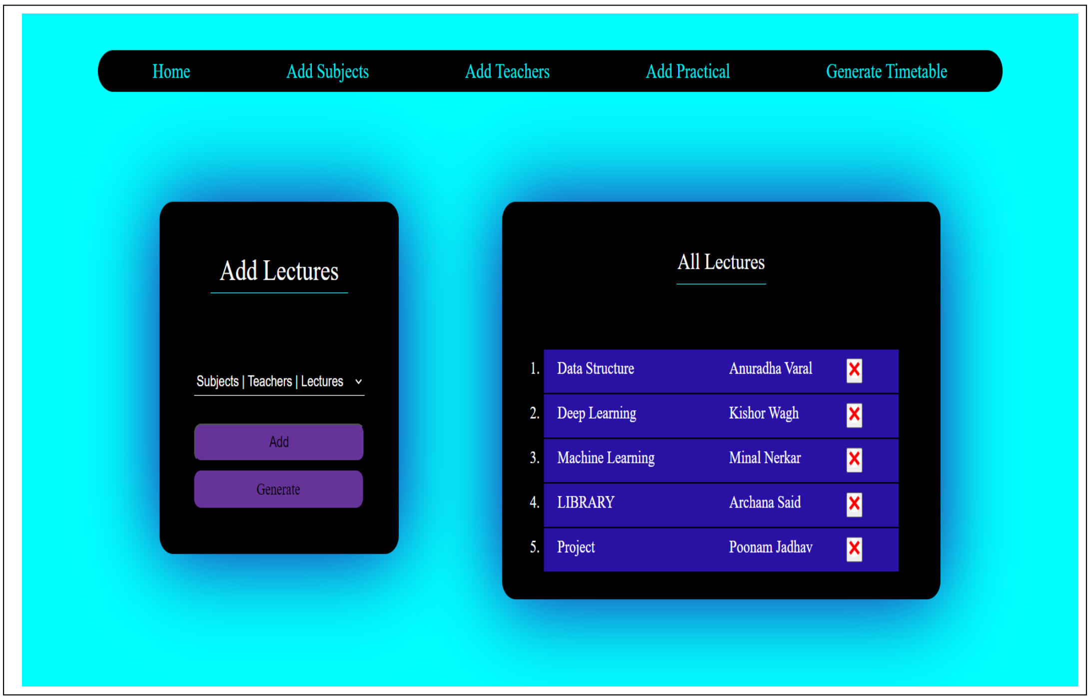
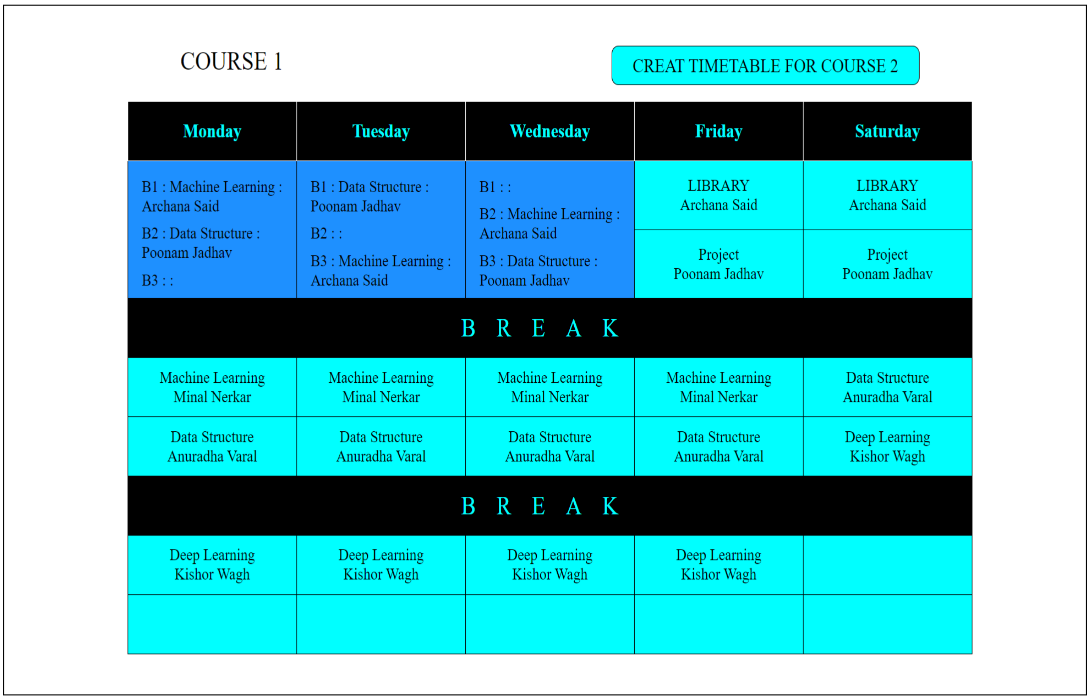


Figure 8.5: Output 5



Monday	Tuesday	Wednesday	Friday	Saturday
B1 : Machine Learning : Archana Said B2 : Data Structure : Poonam Jadhav B3 : :	B1 : Data Structure : Poonam Jadhav B2 : : B3 : Machine Learning : Archana Said	B1 : : B2 : Machine Learning : Archana Said B3 : Data Structure : Poonam Jadhav	LIBRARY Archana Said Project Poonam Jadhav	LIBRARY Archana Said Project Poonam Jadhav
B R E A K				
Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Data Structure Anuradha Varal
Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Deep Learning Kishor Wagh
B R E A K				
Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	

Figure 8.6: Output 6

COURSE 1

Teacher's Timetable

Monday	Tuesday	Wednesday	Friday	Saturday
B1 : Machine Learning : Archana Said B2 : Data Structure : Poonam Jadhav B3 : :	B1 : Data Structure : Poonam Jadhav B2 : : B3 : Machine Learning : Archana Said	B1 : : B2 : Machine Learning : Archana Said B3 : Data Structure : Poonam Jadhav	LIBRARY Archana Said	LIBRARY Archana Said
			Project Poonam Jadhav	Project Poonam Jadhav
B R E A K				
Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Data Structure Anuradha Varal
Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Deep Learning Kishor Wagh
B R E A K				
Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	

COURSE 2

Monday	Tuesday	Wednesday	Friday	Saturday
Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh
Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal
B R E A K				
B1 : Machine Learning : Archana Said B2 : Data Structure : Poonam Jadhav B3 : :	B1 : Data Structure : Poonam Jadhav B2 : : B3 : Machine Learning : Archana Said	B1 : : B2 : Machine Learning : Archana Said B3 : Data Structure : Poonam Jadhav		Machine Learning Minal Nerkar
			Machine Learning Minal Nerkar	Machine Learning Minal Nerkar
B R E A K				
Machine Learning Minal Nerkar	LIBRARY Archana Said	LIBRARY Archana Said	Project Poonam Jadhav	Project Poonam Jadhav

Figure 8.7: Output 7

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

9.1 Conclusion

It is a complicated task to handle Faculty members and allocate lectures to them physically. So our proposed system will help to beat this disadvantage. Thus we can produce a timetable for any number of courses and multiple semesters. This system is user-friendly and provides a faster and better generation of timetable, which in turn saves time and manpower. The project reduces time consumption and the pain in making the timetable manually

9.2 Future work

By using genetic algorithm we are able to reduce the time require to generate time table and generate a timetable which is more accurate, precise and free of human errors. The first phase contains all the common compulsory classes of the institute, which are scheduled by a central team.

9.3 Applications

1. **Colleges**
   A timetable is a kind of schedule that sets out times at which specific events are intended to occur. It may also refer to: colleges timetable, a table for coordinating students, teachers, rooms, and other resources.

2. **Financial firms**
   Financial firms analyze large amount of information every day for decision making. Text summarization can help the analyst to get an idea of market quickly.

3. **Schools**
   A time table ensures that each class has only one teacher during learning hours of a particular period. Time table also reduces the confusion while learning. Students are very clear about the subject they have to study in a particular period.

4. **University**
   University timetable generator involves with scheduling of courses and exams with satisfied faculty requirements, resources and room availability, time slots etc. The timetabling problem has some constraints. While preparing course or exam schedule, lot of constraints arise.

APPENDIX A

1. **Who benefits from this ?**

   1.Administrators: Administrators are the primary beneficiaries of this project. They benefit from the automation of the timetable generation process, which saves them time and effort. The system helps them generate optimized timetables that adhere to various constraints and preferences, resulting in efficient scheduling and resource utilization.

   2.Teachers: Teachers benefit from the project as it ensures a well-organized and balanced timetable. The system takes into account their availability and preferences, allowing for better scheduling and minimizing conflicts..

   3.Students: Students benefit from the project by having a well-designed timetable that avoids conflicts between subjects, reduces overcrowding, and provides a balanced distribution of classes. The system considers their preferences, such as elective subjects or specific time slots, resulting in an improved learning experience.

   4.Support Staff: Support staff, such as lab technicians, librarians, or maintenance personnel, benefit from the project as it ensures that their availability and resource requirements are taken into account during timetable generation.

   5.Institution: The institution as a whole benefits from the project by streamlining the timetable generation process. It reduces the chances of errors or conflicts, optimizes resource allocation, and improves overall efficiency.

2. **What is best/worst case scenario ?**

   Best-Case Scenario:
   In the best-case scenario, the automatic timetable generation process would involve a relatively simple problem with few constraints, preferences, and conflicts. The algorithm or technique used to generate the timetables is highly efficient and capable of quickly finding optimal or near-optimal solutions. The input data, such as teacher availability, room capacities, and student preferences, is accurate and well-defined.
   Worst-Case Scenario:
   In the worst-case scenario, the automatic timetable generation process would

involve a complex problem with numerous constraints, conflicting preferences, and a large search space. The algorithm or technique used may be inefficient or unable to find optimal solutions within a reasonable timeframe. The input data may contain errors, inconsistencies, or ambiguous information, making it challenging to generate feasible timetables.

3. **Where is there most need for this ?**
   1.Large Universities or Colleges: An automatic timetable generator can efficiently handle the complexity and optimize resource utilization, resulting in well-organized and balanced timetables.
   2.Institutions with Multiple Campuses or Branches: Educational institutions with multiple campuses or branches often face the challenge of coordinating and synchronizing timetables across different locations.
   3.Institutions with Rotating or Changing Schedules: Some educational programs or courses may require rotating schedules or periodic changes in timetables. For example, medical schools often have rotating clinical schedules for students.
   4.Schools with Limited Resources: An automatic timetable generator can consider resource capacities, constraints, and availability to allocate resources efficiently, reducing conflicts and maximizing utilization.

4. **Why is this problem/ challenge ?**
   1.Large-Scale Scheduling: Educational institutions often have a large number of courses, teachers, students, and resources that need to be scheduled. The sheer scale of the problem makes manual timetable generation time-consuming and prone to errors. Coordinating the scheduling of multiple courses, assigning teachers to classes, and allocating rooms and resources in an optimized manner is a complex task.
   2.Constraints and Preferences: Timetable generation must take into account various constraints and preferences. These can include teacher availability, room capacities, student preferences for elective courses or time slots, and prerequisites for certain subjects. Balancing and satisfying these constraints and preferences simultaneously while creating the timetable adds complexity to the problem.
   3.Conflicts and Overlaps: Conflicts and overlaps can arise when scheduling courses, teachers, and resources. For example, a teacher may be unavailable during certain time slots, or two courses may require the same room simul-

taneously. Resolving these conflicts and ensuring that classes, teachers, and rooms are scheduled without overlaps is a challenge.

4.Resource Optimization: Efficient utilization of resources, such as classrooms, laboratories, or specialized facilities, is crucial. Timetables need to allocate resources effectively, taking into account their capacities, availability, and compatibility with specific courses. Optimizing resource usage while meeting other constraints requires careful consideration.

5.Dynamic Changes and Updates: Timetables are subject to changes, such as adjustments in course schedules, room availability, or teacher availability. Handling dynamic changes and updates in an efficient and timely manner while maintaining the integrity of the timetable can be challenging. Manual adjustments to accommodate changes may result in cascading conflicts and require significant effort.

6.Preference and Fairness Considerations: Educational institutions often aim to satisfy preferences and ensure fairness in timetable generation. This includes considering student preferences for specific courses or time slots, equitable distribution of classes among teachers, and accommodating specific needs of faculties or programs. Incorporating these considerations while maintaining overall optimization adds complexity to the problem.

APPENDIX B

- **Conference Proceeding :**
  1st Conference on Journal of Emerging Technologies and Innovation Research.

- **Conferene Date :**
  3-5, June 2023

- **Held On :**
  Online Mode

Automatic Timetable Generator Using Genetic Algorithm

Parikshit Shinde
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, Maharashtra, India

Pratik Landghule
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, Maharashtra, India

Archana G. Said
Assistant Professor
Department of Computer Engineering
AISSMS Institute of Information Technology
Pune, Maharashtra, India
archana.said@aissmsioit.org

Pranav Saykar
Department Of Computer Engineering
AISSMS Institute of Information Technology

Tanmay Janrao
Department of Computer Engineering
AISSMS Institute of Information Technology

***Abstract* - While creating a schedule can be challenging, it is crucial in educational institutions. At the moment, the timetable is manually handled. All the periods will be managed automatically, and it will be useful for faculty to have a timetable on their phone via an application. When a teacher is missing, tardy, or early, the schedule will still be managed. For the purpose of efficiently creating a schedule, the maximum workload for a faculty for a day, a week, and a month will be set. It takes a lot of effort and time to make a schedule. In our paper, we attempted to lessen the problems associated with establishing a timetable using a genetic algorithm. In addition to logging into the system, the system administrator also inputs the courses with their codes and the unit The next runner is where all the lecture halls or flats that will be used are entered after the courses have been entered. The system also generates the schedule system after they are entered. This method (inherited algorithm) aids in minimising crimes and errors that may be encountered when creating an automatic timetable. We are able to shorten the time needed to create a timetable and create a schedule that is more exact, precise, and free of fatal offences by employing inheritable algorithms. The institute's common mandatory classes are all included in the first phase and are listed by a central platoon. Individual departmental classes are contained in the alternative phase. Currently, the only way to manually set this schedule is through altering earlier ones.**

I. INTRODUCTION

A timetable is a type of schedule that specifies the times that particular events are supposed to take place. The primary task when an educationalist tries to manually create a schedule is to confirm the work and availability of the faculty members for a particular subject. The creation of a schedule for each course for a given semester while keeping in mind the established university schedule is the other major task. Each subject's time slots should be evenly distributed across a semester such that no faculty member's time slots from one semester overlap with those from another.
The project's proposed approach will make it easier to manage the Schedule without having to repeatedly set it up manually. The system will use machine learning and inputs such semester-by-semester subjects, faculty, and workload of the faculty member.

It will produce a potential timetable for the working days of the week based on these inputs. The proposal is made in "Time Table Scheduling using Genetic Artificial Immune Network." One of the crucial duties that arise in everyday life is scheduling. Many scheduling issues exist, including those involving employees, production, education, and others. Due to the many requirements that must be met in order to find a workable solution, organising educational schedules may also be challenging. Techniques like genetic algorithms have been employed with varying degrees of effectiveness. Scheduling, often known as timetabling, is the practise of allotting time for organised planned actions to produce an outcome that is satisfactory and unrestricted. Transportation, sports, workforce, course, and exam scheduling are some of the use case scenarios.

LITERATURE SURVEY

J.J Grefenstette [1] evolutionary approaches to the time scheduling problem. styles comparable to the evolutionary algorithm set and genetic algorithms. mixed results while in usage. In this essay, we've examined the challenge of using a genetic algorithm to catalogue an instructional timeline. A synthetic inheritable defensive network and a non-original mongrel algorithm were also used to solve the issue, and the outcomes were compared to those obtained using the inheritable algorithm. The findings demonstrate that GAIN is more suitable than GA for arriving at a potential conclusion quickly.

AnujaChowdha [2], Academics continually struggle with the issue of discovering the study schedule that is feasible at the university's major department. This research provides an evolutionary algorithm (EA) technique built on solving the resilient schedule problem at the university. On to chromosomal representation issues. At the correct computer moment, heuristics and contextually grounded thinking may have been attained. Cohesion has been improved by the clever usage of inheritable revision. Using actual data from a top university, the full class plan described in this paper is accepted, estimated, and discussed. An automatic timeline system that.

Anuja Chowdhary [3] developed uses an efficient timing algorithm that can handle both strong and weak barriers. After a specific semester is over, each teacher and student can look at their timetable, but they shouldn't plan ahead. According to the teacher's schedule, the availability and power of visual resources, and other rules relevant to various courses, semesters, preceptors, and grade positions, the Timetable Generator System develops a timeline for each class and teacher.

Anirudha Nanda [4] proposes a typical resolution to the timing issue. most difficult past heuristic programmes that have been suggested from a scholarly perspective. This outcome is still valid when viewed from the perspective of the topic, specifically the teacher's inadequacy at a certain moment. The planning result described in this work is adaptable, with the main goal of resolving academic and academic dispute, schoolteacher-related concerns. All implicit walls (e.g., schoolteacher vacuity, etc.) are replied firmly.

Al-Khair [5] suggested using algorithmic techniques to solve the timing issue with teacher availability admissions. The challenge of scheduling academy time is completely addressed by this technique, which employs a heuristic approach. First, it creates a temporary timeline using randomly generated title sequences. If a teacher divides a class into more subjects than are allowed, the extra subjects are moved to the Conflict data structure. By taking into account the tight relationship between the schedule problem and the vertex colouring problem,

Csima and Gotlieb [6] also addressed the schedule problem as a three-dimensional assignment problem. The now well-known particular link between the diverse scheduling difficulties was originally examined in this study. Becker deceived their respective executions with "hand" computations, extending the work These publications typically relied on a heuristic methodology. This work led to a large number of papers that examined the issue but didn't add any fresh insight..

Broder [7] asserted that the objective function can be minimised by repeatedly assessing a collection of relevant nonlinear equations (deduced from arbitrary or montecarlo assignments). It was asserted that the outcomes thus obtained might not be ideal but rather would be comparable to the initial minimal outcomes obtained by prior heuristic methods.

Chan H [8] as being crucial for the functionality of Gotlieb's set partitioning system of schedule generation. However, the first algorithm itself experienced duplicated effectiveness as a result of this. Lions added a conception system for the creation of class/teacher calendars to this basic task. In his paper, he discussed how to apply Kuhn's Hungarian system to the matrix reduction issue needed by Gotlieb's schedule generation method.

Walker and Macon [9] They implemented a Monte Carlo algorithm that randomly chooses classes for students, illustrating the difficulty of placing students in a set of courses. They observed that the application of the randomising procedure had a limit on the ultimate best outcome.

Punter [10] His method entailed transferring this data onto rendering wastes before it was processed by a computer, as well as manually producing allocation plans that outlined the instructional circumstances for each class. The discovery of contradictory sub-sets of assignment conditions was made possible by analysis employing a graph colouring approach. It was claimed that the system might be used to generate workable calendars with the appropriate emendations

Hulskamp [11], who created software to assist engineering students in investigating a wide range of potential solutions to a design problem. The software was put into use at Caulfield Institute of Technology, and it was successful in illuminating the problem's complexity and phase space. This may be the first publication of a DSS for the creation of schedule results.

Deb K [12] created a programme that was not based on integer programming but was also effective and effective. As a DSS, it made it simple for users and the programme to collaborate on finding the best solution. Calendars were successfully constructed (as opposed to generated) according to empirical tests.

ALGORITHM WORKING

Genetic Algorithm-The inheritable algorithm, which is based on natural selection, the process that drives natural elaboration, is a method for solving both limited and unconstrained optimization problems. A population of individual results is continuously modified by the inheritable algorithm.

Algorithm Working-

Steps Involved-

1.Initial Population-

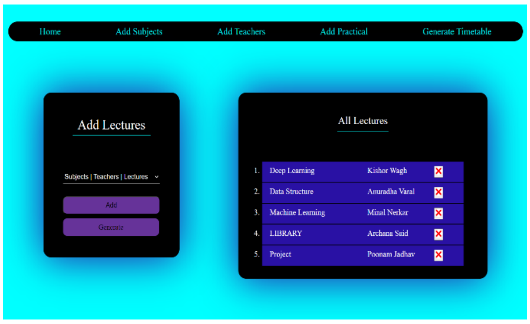
Initially, we were supposed to create the first-year class schedule by entering all of the section-by-section information. Read the database's faculty, subjects, and section information. Here algorithm will generate first possible solution by randomly assessing inputs known as chromosomes.

2.Fitness Function-

In this step we will consider all the inputs such as teachers classrooms timeslots and students and assign them fitness value as and try to arrange them such as there is no clash between various inputs like there might be some extra classrooms which are available thoroughly during college session we can accommodate those classes to any particular timeslot. Fitness function plays very crucial role as it is a building block on which our algorithm is going to rely on.

3.Selection-

Selection step selects the best possible solution containing students teachers subjects classrooms and timeslots and the Chromosomes within the population are chosen by this driver for reduplication. It is more likely to be chosen for breeding over time the better the chromosomal fit. A fraction of the accessible population is chosen to elect a new generation for each succeeding product. A fitness-based approach is used to select each individual result, with fitter results often having a higher likelihood of selection.

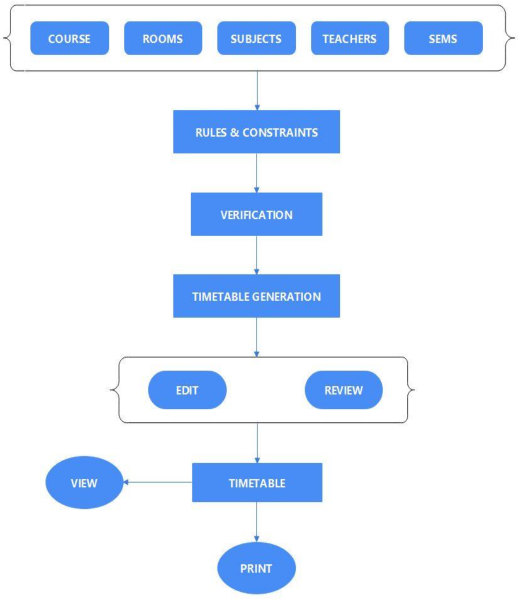


4.Crossover-

Crossover takes the best possible solutions produced by the step 3 but there might be still some clashes between some constraints like time slot overlapping or assigning same classroom for two different subjects at same time also sometimes same teacher could be assigned to different classes at same time we solve this problem in this step.

5.Mutation-

Mutations introduce randomness in the results and it maintains diversity ultimately resulting in avoidance of premature termination like sometimes teacher might not want to teach in particular class also some professors might prefer larger classrooms inspite of the smaller number of students mutation takes cares of such situations results might not always be perfect.



ARCHITECTURE OF THE SYSTEM

The system's capacity to input the different courses, lecture halls, departments, programmes, lecturers, and the definition of a few limits from which the timetable is built reduces the high cost and lengthy turnaround necessary in the creation of close to ideal timetables.

Timetabling genetic algorithm approach scheduling a Timetable for college –

As mentioned, scheduling classes for a council schedule is constantly encountered With constraints(Hard and Soft) due to diversity as compared to a academe schedule where conditions are largely limited. The problems associated with the hard constraints need to be resolved to produce a functional result. To optimize the performance of the of the scheduling it is important to address the issues linked to soft constraints. Still, a careful approach should be formulated without compromising the affect to hard constraints to minimize serious disruptions to the system. As analogous, a straightforward scheduling system as for without switching to a different strategy, a tiny academic system cannot be applied to a complicated association in a way that produces quick and effective outcome. Several soft restrictions may clash and as a result, a compromise will need to be made between them. The soft restriction will assign an appropriate classroom that can fit the students. For instance, a class might have 12 students. Yet, the lecturer might prefer a room with capacity of the 45 students. As a professor choice is taken into account in the soft constraints, it is hoped that the class scheduler will identify a proffered configuration

The following information will serve as the foundation for the class scheduling issue.-

available academics accessible cells Schedules for student groups There may be free time intervals on a council student's timetable, which sets it apart from a grade-academe schedule. This is based on how many classes the scholars take. The class scheduler will designate a time slot, a teacher, a location, and a student group for each class. By multiplying the number of student groups by the number of modules each student group is registered in, it is possible to calculate the total number of classes that need to be planned. The following hard limitations will be taken into account for every class schedule produced by this process.

Only free classrooms allow for the listing of classes. A professor is only able to instruct one class at a time. The student group must fit in the classroom, thus it must be large enough. A particular number of class packets are required while coding the class schedule. The class's scheduled time, the instructor's instruction of the group, and the necessary classroom are the classes

time space niche lecture course or module RoomID, Room Number, Room Capacity, Course/ ModuleID, Course/ Module Code (Sections), and Course/ Module Timeslot ID and Timeslot Professor Name and Professor ID. ID, Professor. Group ModuleID, GroupID, and Group Size CClass Denotes the student's current class. IDs for the class, group, module, professor, timeslot, and room. All of these items will be recapped in the timetable. This class will be useful for separating the interactions between various restrictions. Additionally, it will be in charge of parsing chromosome for use as an example in an instruction algorithm to create a candidate timetable that can be estimated and graded.

COURSE 1

Teacher's Timetable

Monday	Tuesday	Wednesday	Friday	Saturday
B1 : Machine Learning : Archana Said B2 : Data Structure : Poonam Jadhav B3 : :	B1 : Data Structure : Poonam Jadhav B2 : : B3 : Machine Learning : Archana Said	B1 : : B2 : Machine Learning : Archana Said B3 : Data Structure : Poonam Jadhav	LIBRARY Archana Said	LIBRARY Archana Said
			Project Poonam Jadhav	Project Poonam Jadhav
B R E A K				
Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Machine Learning Minal Nerkar	Data Structure Anuradha Varal
Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Deep Learning Kishor Wagh
B R E A K				
Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	

COURSE 2

Monday	Tuesday	Wednesday	Friday	Saturday
Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh	Deep Learning Kishor Wagh
Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal	Data Structure Anuradha Varal
B R E A K				
B1 : Machine Learning : Archana Said B2 : Data Structure : Poonam Jadhav B3 : :	B1 : Data Structure : Poonam Jadhav B2 : : B3 : Machine Learning : Archana Said	B1 : : B2 : Machine Learning : Archana Said B3 : Data Structure : Poonam Jadhav		Machine Learning Minal Nerkar
			Machine Learning Minal Nerkar	Machine Learning Minal Nerkar
B R E A K				
Machine Learning Minal Nerkar	LIBRARY Archana Said	LIBRARY Archana Said	Project Poonam Jadhav	Project Poonam Jadhav

CONCLUSION

Handling faculty members and physically allocating lectures to them is a difficult task. So, our suggested system will aid in overcoming this drawback. As a result, we are able to create a schedule for any number of courses over several semesters. This system is simple to use and offers quick and efficient schedule generation, saving time and resources. The layout lessens the time commitment and discomfort of manually creating the timetable.

REFERENCES

[1] J.J Grefenstette, editor. Proceedings of the First International Conference on Genetic Algorithms and their Applications. Practice and Theory of Automated Timetabling VI Proceedings of The 6th International Conference on the practice and Theory of Automa.

[2] AnujaChowdha ry, PriyankaKakde, ShrutiDhoke, SonaliIngle,RupalRushiya, Di- nesh Gawande, 'Time table Generation System", International Journal of Comp.

[3] Anuja Chowdhary A. White Paper: "A hybrid GA-heuristic search strategy." AI Expert,USA.

[4] Anirudha Nanda, Marco Dorigo, Vittorio Manniezzo (1992). "A Genetic Algo- rithm to Solve the Timetable Problem" Journal of Computational Optimization and Applications, 1, 90-92.

[5] Al-Khair ,Fischer T., Gubbels H., Hacker C., Hasprich O., Scheibel C., We- icker K., Weicker N., Wenig M., Wolfangel C. Automated solution of a highly constrained school timetabling problem - preliminary results. EvoWorkshops, Como-Italy.

[6] Csima and Gotlieb ,Elliman D and Weare R ."A genetic algorithm for university timetabling system." Presented at the East-West Conference on Computer Technologies in Education, Crimea, Ukraine.

[7] Broder ,A multiobjective genetic algorithm for the class/teacher timetabling problem." In Proceedings of the Practice and Theory of Automated Timetabling (PATAT'00), Lecture Notes in Computer Science,Springer, 2079, 3-17.

[8] Chan H. W."School Timetabling Using GeneticSearch." 2th International Con- ference on thePractice and Theory of Automated Timetabling, PATAT'97.

[9]Walker and Macon"An updated survey of GA- based multiobjective optimization tech-niques." ACMComputing Surveys, 32(2), 109-143.

[10]Punter"A tabu search algorithm for computing an operational timetable." European Journal of Operational Research, 76(1), 98-110.

[11] Hulskamp., Deb K., Fonseca, C.M. (2006). Multi-objective evolutionary algorithm for university class timetabling problem, In Evolutionary Scheduling, Springer- Verlag Press.

[12] Deb K., Agarwal S., Pratap A., Meyarivan T. (2002). "A fast and elitist multi- objectivegenetic algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation,6(2), 182-197.

DON BOSCO COLLEGE OF ENGINEERING

FATORDA - GOA

Certificate of Appreciation

Presented to

Mrs.Archana.G.Said

guide for the project titled

AUTOMATIC TIMETABLE GENERATOR USING GENETIC ALGORITHM

for the guidance and participation in "Technix 2023"
a National Level Project Competition
organised by Final Year Project Quality Assurance Committee
in association with Don Bosco Electronics and Telecommunication Society (DBEATS)
on May 13th, 2023.

Dr. D.S. Vidhya (H.O.D. - ETC)	Dr. Neena S. Panandikar (Principal)	Fr. Kinley D'Cruz (Director)

DON BOSCO COLLEGE OF ENGINEERING

FATORDA - GOA

Certificate

This is to certify that

PRANAV SAYKAR

participated and was shortilsted for the Final Phase of "Technix 2023"
a National Level Project Competition
organised by Final Year Project Quality Assurance Committee
in association with Don Bosco Electronics and Telecommunication Society (DBEATS)
on May 13th, 2023.

Dr. D.S. Vidhya (H.O.D. - ETC)	Dr. Neena S. Panandikar (Principal)	Fr. Kinley D'Cruz (Director)

DON BOSCO COLLEGE OF ENGINEERING

FATORDA - GOA

Certificate

This is to certify that

PARKSHIT SHINDE

participated and was shortilsted for the Final Phase of "Technix 2023"
a National Level Project Competition
organised by Final Year Project Quality Assurance Committee
in association with Don Bosco Electronics and Telecommunication Society (DBEATS)
on May 13th, 2023.

Dr. D.S. Vidhya (H.O.D. - ETC)	Dr. Neena S. Panandikar (Principal)	Fr. Kinley D'Cruz (Director)

DON BOSCO COLLEGE OF ENGINEERING

FATORDA - GOA

Certificate

This is to certify that

TANMAY JANGRAO

participated and was shortilsted for the Final Phase of "Technix 2023"
a National Level Project Competition
organised by Final Year Project Quality Assurance Committee
in association with Don Bosco Electronics and Telecommunication Society (DBEATS)
on May 13th, 2023.

Dr. D.S. Vidhya (H.O.D. - ETC)	Dr. Neena S. Panandikar (Principal)	Fr. Kinley D'Cruz (Director)

DON BOSCO COLLEGE OF ENGINEERING

FATORDA - GOA

Certificate

This is to certify that

PRATIK LANDGHULE

participated and was shortilsted for the Final Phase of "Technix 2023"
a National Level Project Competition
organised by Final Year Project Quality Assurance Committee
in association with Don Bosco Electronics and Telecommunication Society (DBEATS)
on May 13th, 2023.

Dr. D.S. Vidhya (H.O.D. - ETC)	Dr. Neena S. Panandikar (Principal)	Fr. Kinley D'Cruz (Director)

APPENDIX C: PLAGIARISM

Ouriginal
by Turnitin

Document Information

Analyzed document	Project_BlackBook.pdf (D168914991)
Submitted	5/30/2023 7:32:00 AM
Submitted by	Pushpa V.Hirave
Submitter email	aissmsioit.library@aissmsioit.org
Similarity	0%
Analysis address	aissmsioit.library.aissms@analysis.urkund.com

References

[1] [1] J.J Grefenstette, editor. Proceedings of the First International Conference on Genetic Algorithms and their Applications. Practice and Theory of Automated Timetabling VI Proceedings of The 6th Internet 10.1145/2222444.2222467.

[2] [2] AnujaChowdha ry, PriyankaKakde, ShrutiDhoke, SonaliIngle,RupalRushiya, Di- nesh Gawande, 'Time table Generation System", International Journal of Comp.1289- 1294, doi: 10.1109/ICOEI.2019.8862553.

[3] Anuja Chowdhary A. White Paper: "A hybrid GA heuristic search strategy." AI Expert,USA.

[4] Anirudha Nanda, Marco Dorigo, Vittorio Manniezzo (1992). "A Genetic Algorithm to Solve the Timetable Problem" Journal of Computational Optimization and Applications, 1, 90-92

[5] Al-Khair ,Fischer T., Gubbels H., Hacker C., Hasprich O., Scheibel C., Weicker K., Weicker N., Wenig M., Wolfangel C. Automated solution of a highly constrained school timetabling problem - preliminary results. EvoWorkshops, Como-Italy,pp. 1-6, doi: 10.1109/SCEECS.2014.6804439.

[6] Csima and Gotlieb ,Elliman D and Weare R ."A genetic algorithm for university timetabling system." Presented at the East-West Conference on Computer Technologies in Education, Crimea, Ukraine. doi: 10.1109/ICCUBEA.2018.8697465.

[7] Broder ,A multiobjective genetic algorithm for the class/teacher timetabling problem." In Proceedings of the Practice and Theory of Automated Timetabling (PATAT'00), Lecture Notes in Computer Science,Springer, 2079, 3-17. : 10.1109/ICCIDS.2019.8862030.

[8] Chan H. W."School Timetabling Using GeneticSearch." 2th International Con- ference on thePractice and Theory of Automated Timetabling, PATAT'97.

[9] Walker and Macon"An updated survey of GA- based multiobjective optimization tech-niques." ACMComputing Surveys, 32(2), 109-143. doi: 10.1109/ICDIM.2018.8847156.

[10] Punter"A tabu search algorithm for computing an operational timetable." European Journal of Operational Research, 76(1), 98-110. doi: 10.1109/ICICT46931.2019.8977708.

[11] Hulskamp., Deb K., Fonseca, C.M. (2006). Multiobjective evolutionary algorithm for university class timetabling problem, In Evolutionary Scheduling, Springer- Verlag Press. doi: 10.1109/YAC.2019.8787615.

[12] Deb K., Agarwal S., Pratap A., Meyarivan T. (2002). "A fast and elitist multi-objectivegenetic algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation,6(2), 182-197. , doi: 10.1109/TSSA.2018.8708744.

[13] Wright M (1996). "School timetabling using heuristic search." Journal of Oper- ationalResearch Society., doi: 10.1109/ICADIWT.2014.6814680.

[14] Hemmerling "Ant Algorithms for the Exam Timetabling Problem." 6th International Conference on the Practice and Theory of Automated Timetabling, PATAT'06.. pp. 515-521, doi: 10.1109/WI.2016.0087.

[15] Knauer . "The construction of class-teacher timetables." Proceedings of IFIP Congress, North-Holland Pub. Co., Amsterdam., doi: 10.1109/ICIS46139.2019.8940293.

AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
ADDING VALUE TO ENGINEERING

Department of Computer Engineering

Self-Evaluation

Name of the Project	Text Summarization using TF-IDF and Textrank algorithm
Name of the student	Pranav Saykar
Name of the student	Pratik Landghule
Name of the student	Parikshit Shinde
Name of the student	Tanmay Janrao

File of literature Survey	Design	Implementation	Task and Result	Attendance on project Day	Work according to plan Activity	Managing of log book	Project Presentation or Participation	Project exhibition participation	Award Prize if any
(5)	(20)	(20)	(20)	(5)	(10)	(5)	(5)	(5)	(5)

Observations and comments of Guide:

Signature of Guide

Name of Student **Signature of student**

1. Pranav Saykar

2. Pratik Landghule

3. Parikshit Shinde

4. Tanmay Janrao